



Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Desarrollo de Aplicaciones Web 2.0



---

The Geek Store

---



Alumno: Cesar Rebollar Mondragón

Profesora: Angie Aguilar Domínguez

Fecha: 28/Mayo/2019

Grupo: 2892

# Contenido

1. Introducción.....	3
2. Requerimientos de instalación .....	4
3. Instalación y configuración .....	4
4. Funcionamiento del sistema .....	6
a. Caso de uso:.....	6
b. Diagrama Entidad-Relación:.....	7
c. Especificaciones de Caso de Uso: .....	8
5. Diccionario de datos y DER de la base de datos.....	11
6. Conclusión .....	13
7. Enlaces consultados .....	14
8. Anexos .....	16

# 1. Introducción

En el presente proyecto tiene como objetivo realizar una aplicación web para la venta de artículos en línea desde la cual, una empresa o negocio pueda ampliar sus ventas mediante el comercio electrónico de una manera rápida y sencilla. De esta manera, los clientes que hayan realizado un previo registro podrán consultar los productos de su preferencia y realizar compras con distintas formas de pago sin la necesidad de salir de casa.

Por otra parte, al administrador podrá agregar nuevos productos, modificar los existentes y borrar los que ya no se necesiten, tanto de productos como de las ventas realizadas en la base de datos, además de la corrección de datos erróneos que puedan aparecer en el registro de algún cliente.

Para comenzar con esta tarea, decidí tomar la experiencia de algunas páginas web que utilizan el mecanismo de venta online para tratar de estudiar y analizar las funcionalidades más importantes que se deberían desarrollar. Tal es el caso de kichink.

La tienda en línea lleva como nombre “The Geek Store”, concepto que se basa en la venta de cómics de las compañías Marvel y DC. Motivo por el cual, en los últimos años se ha disparado el auge de los superhéroes a nivel internacional y mundial. Por lo tanto, se estima que gran parte de usuarios tengan la curiosidad de entrar y visualizar el contenido que se encuentre en tendencia.

En cuanto al desarrollo de esta aplicación web, está realizada mediante programación en PHP, plantillas CSS y Bootstrap. Además, con la creación de una base de datos en PostgreSQL, permitirá una mejor gestión de los datos en la tienda en línea ya que ofrece una gran cantidad de opciones avanzadas y también una ventaja en cuanto rendimiento en cada transacción de venta que se realice.

Todo lo anterior, está basado en un modelo de Virtual Host que nos permite asignar distintos sitios web alojados en un servidor apache a través de una misma dirección IP. Además de contar con un repositorio en GitHub y si es posible integrarlo con Paypal.

## 2. Requerimientos de instalación

Una vez desarrollada la aplicación, se necesitan los siguientes requerimientos de instalación para su correcto funcionamiento:

- Sistema Operativo Debian 9.4.9.
- Ram 2 Gb.
- Disco duro 20 Gb
- Virtual Host Apache HTTP Server Versión 2.4.
- PHP versión 7.0.33.
- Apache versión 2.4.25.
- PostgreSQL versión 9.6.13.

## 3. Instalación y configuración

A la hora de implementar la configuración de la aplicación web, se ha destinado la siguiente ruta de instalación del proyecto:

- /var/www/html/ProyectoFinalWebShop

**Dentro de la carpeta ProyectoFinalWebShop, la estructura es la siguiente:**

- **/ProyectoFinalWebShop:**
  - Carpeta donde se encuentran la mayoría de archivos para el correcto funcionamiento de la tienda en línea. Archivo inicial “index.php”.
    - **/Admin:**
      - Carpeta que contiene archivos para gestionar los productos.
    - **/Archivos:**
      - Carpeta que contiene los archivos de registro e ingreso del usuario mediante la utilización de sesiones. Así como, consultas y registros en la base de datos.
    - **/Global:**
      - Carpeta que contiene archivos de conexión de la base de datos.
    - **/Images**
      - Carpeta que contiene imágenes utilizadas en la aplicación web.

- **/Templates**

- Carpeta que contiene la estructura de la cabecera y pie de la página web. Si requiere modificaciones, estas serán más fáciles de realizar al estar separadas del resto del código.

**Se configuraron los siguientes archivos para el funcionamiento del Virtual Host:**

- **Dentro de la carpeta /etc/apache2/sites-available/:**
  - Se realizó una copia del archivo 000-default.conf con nombre “webshop.conf.”
  - Se agregaron las siguientes líneas:
    - ServerAdmin <nombre>@localhost.local
    - ServerName www.thegeekshop.com.mx
    - DocumentRoot /var/www/ProyectoFinalWebShop
    - ErrorDocument 404 "Recurso no disponible"
- **Dentro de la carpeta /etc/:**
  - Se agregó la siguiente línea en el archivo “hosts”:
    - 127.0.0.1     www.thegeekshop.com.mx

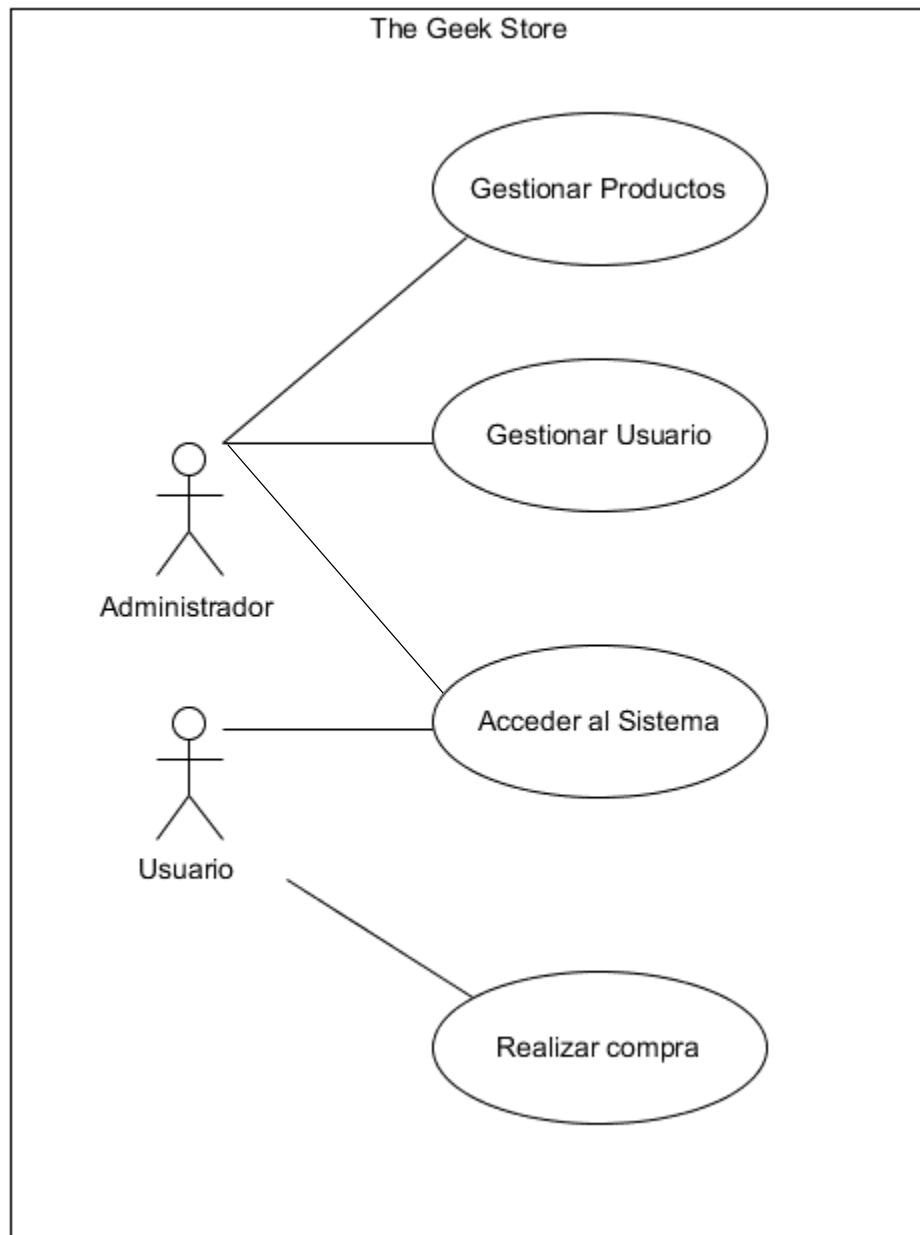
**Para la configuración de la base de datos:**

- Se define el usuario: “admin\_shop” y “usuario”.
  - **admin\_shop:**
    - Usuario con privilegios Insert, Update, Delete.
  - **usuario:**
    - Usuario con privilegios Select, es decir, solo podrá consultar los productos de la base de datos.
- Se procede a la creación de la base de datos: “geekshop”
- Una vez configurados los archivos de instalación, se procede a arrancar el servidor con el comando service postgresql start y mediante un navegador web, se accede al siguiente link: [www.thegeekshop.com.mx](http://www.thegeekshop.com.mx).

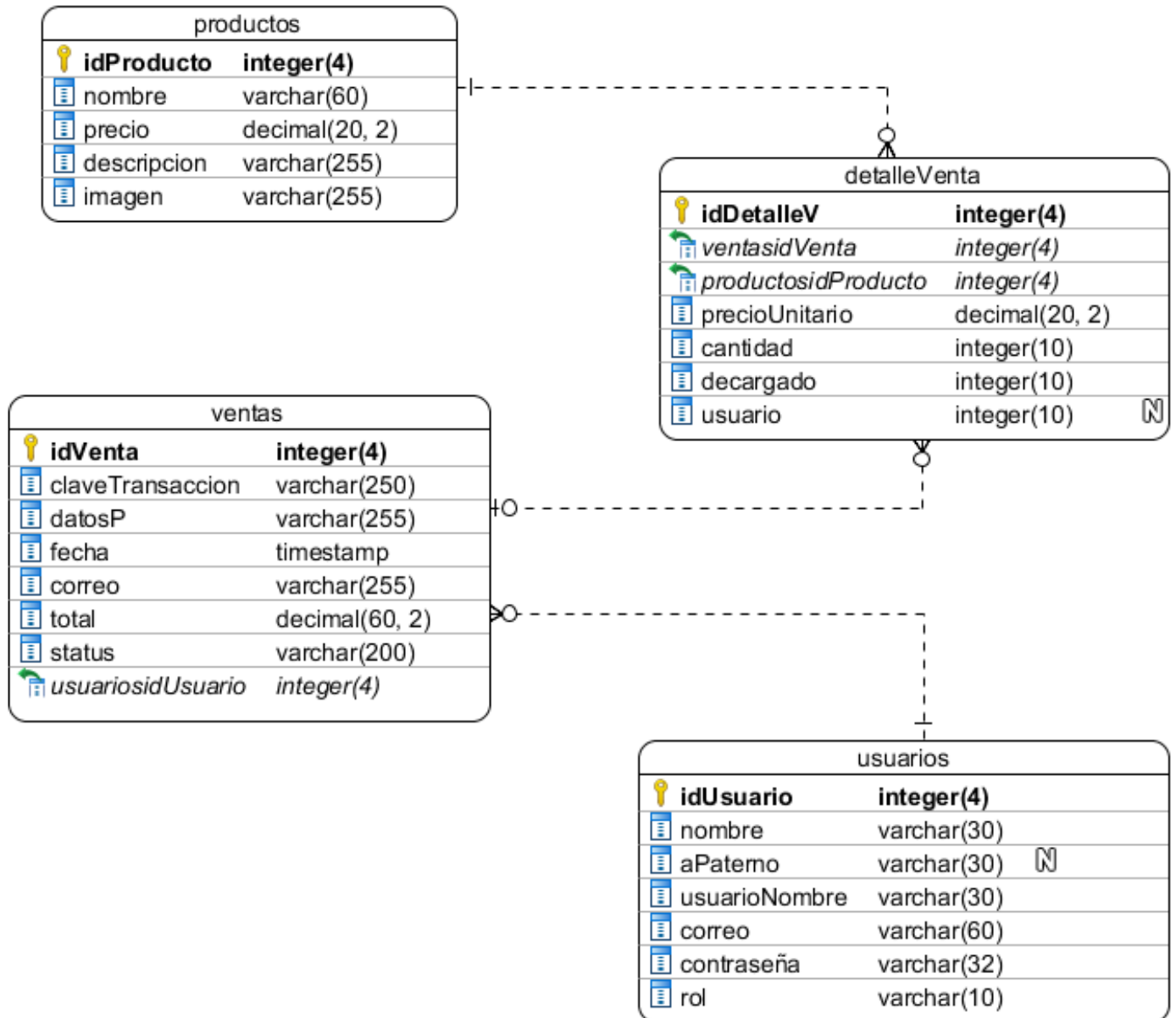
**Nota:** Para más información, ir “8. Anexos”.

## 4. Funcionamiento del sistema

### a. Caso de uso:



## b. Diagrama Entidad-Relación:



### c. Especificaciones de Caso de Uso:

Nombre C.U	Acceso al sistema
<b>Actores</b>	Usuario sin privilegios y Administrador del sistema encargados de proporcionar los datos necesarios para el inicio de sesión.
<b>Descripción</b>	Permite la autenticación a los usuarios para que puedan hacer uso de la aplicación web.
<b>Precondiciones</b>	El Usuario y Administrador del sistema deben de contar con un nombre de usuario y una contraseña para poder ingresar a la aplicación web.
<b>Flujo básico y alternativo</b>	<p>El caso de uso inicia cuando el usuario requiere ingresar al sistema:</p> <ol style="list-style-type: none"> <li>1. El sistema solicita los siguientes datos: Nombre de usuario y contraseña y muestra la opción "Iniciar Sesión".</li> <li>2. El usuario proporciona los datos y selecciona iniciar sesión.</li> <li>3. El sistema verifica si los datos proporcionados son correctos. <b>E1</b> <ol style="list-style-type: none"> <li><b>3.1. Si el usuario es administrador del sistema:</b> <ol style="list-style-type: none"> <li>3.1.1. El sistema muestra las opciones de: Agregar Producto, Agregar Usuarios y Cerrar Sesión.</li> </ol> </li> <li><b>3.2. Si es Usuario sin privilegios:</b> <ol style="list-style-type: none"> <li>3.2.1. El sistema muestra las opciones: Inicio, Ver Carrito, Cerrar Sesión.</li> </ol> </li> </ol> </li> <li>4. El caso de uso termina.</li> </ol>
<b>Poscondiciones</b>	Ninguna.
<b>Requerimientos especiales</b>	Ninguna.
<b>Excepciones</b>	<b>E1:</b> Si el nombre de usuario o contraseña no es válido, el sistema manda un mensaje "Usuario o Contraseña incorrecto". La aplicación web solicita ingresar nuevamente los datos.

Nombre C.U	Gestionar Productos
<b>Actores</b>	Administrador del sistema encargado de proporcionar los datos necesarios para realizar el registro, modificación y eliminación de un producto.
<b>Descripción</b>	Este caso de uso permite el registro, modificación y eliminación de un producto.
<b>Precondiciones</b>	El Administrador del sistema debe de haber iniciado sesión a la aplicación web previamente.
<b>Flujo básico y alternativo</b>	<p>El caso de uso inicia cuando se necesita registrar, modificar y eliminar los datos de algún producto:</p> <ol style="list-style-type: none"> <li>1. El Administrador del sistema selecciona la opción: Agregar Productos.</li> <li>2. El sistema muestra las opciones: Agregar, Modificar Eliminar y Cancelar.</li> <li><b>3. Si el Administrador selecciona Agregar:</b> <ol style="list-style-type: none"> <li>3.1. El sistema solicita los siguientes datos: Nombre del producto, Precio, Descripción y Foto del producto.</li> </ol> </li> </ol>



	<p>3.2. El usuario proporciona los datos solicitados y selecciona la opción Agregar. <b>E1</b></p> <p>3.3. El sistema verifica los datos.</p> <p>3.4. Si los datos son correctos:</p> <p>3.4.1. El sistema notifica: "Registro realizado exitosamente".</p> <p>3.4.2. El flujo termina.</p> <p><b>4. Si el Administrador selecciona Modificar:</b></p> <p>4.1. El sistema muestra una lista de los productos registrados.</p> <p>4.2. El administrador selecciona un producto.</p> <p>4.3. El sistema muestra el Nombre del producto, Precio, Descripción y Foto del producto y la opción modificar.</p> <p>4.4. El Administrador modifica los datos del producto.</p> <p>4.5. El Administrador selecciona la opción Modificar.</p> <p>4.6. El sistema verifica que los datos sean válidos. <b>E1</b></p> <p>4.7. El sistema modifica los datos.</p> <p>4.8. El flujo termina.</p> <p><b>5. Si el Administrador selecciona Eliminar:</b></p> <p>5.1. El sistema muestra una lista de los productos registrados.</p> <p>5.2. El administrador selecciona un producto.</p> <p>5.3. El sistema muestra el Nombre del producto, Precio, Descripción y Foto del producto y la opción Eliminar.</p> <p>5.4. El sistema notifica: "¿Está seguro de borrar el producto?" y muestra las siguientes opciones: Sí, Cancelar.</p> <p>5.4.1. Si el usuario selecciona Sí.</p> <p>5.4.1.1. El sistema borra el producto.</p> <p>5.4.2. Si el usuario selecciona Cancelar.</p> <p>5.4.3. El caso de uso se reinicia.</p> <p>5.4.4. El flujo termina.</p> <p><b>6. Si el Administrador selecciona Cancelar:</b></p> <p>6.1. El sistema muestra una lista de los productos registrados.</p> <p>6.2. El administrador selecciona un producto.</p> <p>6.3. El sistema muestra el Nombre del producto, Precio, Descripción y Foto del producto y la opción Cancelar.</p> <p>6.4. Si el Administrador selecciona la opción Cancelar.</p> <p>6.4.1. El caso de uso se reinicia.</p> <p>6.4.2. El flujo termina.</p> <p>7. El caso de uso Gestionar Productos termina.</p>
<b>Poscondiciones</b>	<p>Agregar datos del producto.</p> <p>Modificar datos del producto.</p> <p>Eliminar datos del producto</p>
<b>Requerimientos especiales</b>	Ninguna.

<b>Excepciones</b>	<b>E1.</b> Todos los campos son obligatorios. La aplicación web manda un mensaje: “Llenar todos los campos”. La aplicación permite ingresar los datos nuevamente.
--------------------	---

<b>Nombre C.U</b>	<b>Realizar Compra</b>
<b>Actores</b>	Usuario sin privilegios encargado de proporcionar los datos necesarios para realizar una compra.
<b>Descripción</b>	Permite la compra de productos a través de un carrito de compras.
<b>Precondiciones</b>	El Usuario deben de contar con un nombre de usuario y una contraseña para poder ingresar a la aplicación web.
<b>Flujo básico y alternativo</b>	<p>El caso de uso inicia cuando el usuario desea comprar algún producto:</p> <ol style="list-style-type: none"> <li>1. El sistema muestra una lista de productos y muestra la opción “Iniciar Sesión”.</li> <li>2. El usuario proporciona los datos: Nombre de Usuario y Contraseña y selecciona iniciar sesión.</li> <li>3. El sistema muestra una lista de productos y muestra la opción “Agregar a Carrito”.</li> <li>4. El sistema agrega los productos al carrito y muestra la opción “Ver carrito”.</li> </ol> <p><b>4.1. Si el usuario selecciona “Ver carrito”:</b></p> <ol style="list-style-type: none"> <li>4.1.1. El sistema muestra una lista de los productos agregados por el usuario y la opción de “Proceder a Pagar”.</li> <li>4.1.2. El sistema muestra una lista de opciones de pago.</li> <li>4.1.3. El usuario selecciona una opción de pago.</li> <li>4.1.4. Si el pago fue correcto: <ol style="list-style-type: none"> <li>4.1.5. El sistema muestra un mensaje en pantalla “Listo, pago exitoso”.</li> <li>4.1.6. El sistema muestra una lista de los productos adquiridos por el usuario y la opción “Descargar”.</li> </ol> </li> <li>5. El caso de uso termina.</li> </ol>
<b>Poscondiciones</b>	Ninguna.
<b>Requerimientos especiales</b>	Ninguna.
<b>Excepciones</b>	Ninguna.

## 5. Diccionario de datos y DER de la base de datos.

**Tabla: usuarios**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Descripción</b>
idUsuario	integer	04	Clave única del usuario.
nombre	varchar	30	Nombre del usuario.
aPaterno	varchar	30	Apellido del usuario.
usuarioNombre	varchar	30	Alias del usuario.
correo	varchar	60	Correo del usuario para registro.
contraseña	varchar	32	Contraseña del usuario para acceder al sistema.
idTipoUsuario	integer	04	Llave foránea de la tabla tipoUsuario que indica el tipo de usuario.
Rol	Varchar	10	Rol del usuario, si es administrador o usuario normal.

**Tabla: productos**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Descripción</b>
idProducto	integer	04	Clave única del producto.
nombre	varchar	60	Nombre del producto.
precio	decimal	20,2	Precio actual del producto.
descripcion	varchar	255	Descripción del producto.
imagen	varchar	255	Imagen del producto.

**Tabla: ventas**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Descripción</b>
idVenta	integer	04	Clave única de la venta realizada.
claveTransaccion	varchar	250	Transacción de la venta en ese momento de la sesión del carrito.
datos	varchar	255	Datos que contiene el array de PayPal.
fecha	timestamp	0	La fecha en que se realizó la venta.
correo	varchar	255	Correo del usuario que está comprando el producto.
total	decimal	60,2	El total de los productos que está comprando el usuario.

status	varchar	200	El estado de la venta realizada. Si el usuario pagó el producto, se marcará el estado como “completado”, de lo contrario seguirá en “pendiente”.
--------	---------	-----	--

**Tabla: detalleVenta**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Descripción</b>
idDetalleV	integer	04	Clave única del detalle de la venta. Se registra la venta realizada por el cliente por cada producto comprado.
idVenta	integer	04	Clave única de la venta realizada.
idProducto	integer	04	Clave única del producto.
precioUnitario	decimal	20,2	Precio unitario por cada producto comprado.
cantidad	integer	10	La cantidad que el usuario compró del producto.
descargado	integer	10	Campo que registra si el usuario ha descargado su compra.

## 6. Conclusión

Al realizar un proyecto de esta magnitud, pude darme cuenta que se necesita de gran cantidad de conocimientos, tanto de requerimientos como de programación para llevarlo a cabo. Este proyecto es un claro ejemplo de que hay que estar más preparados a la hora de enfrentarnos al mercado laboral una vez finalizada la carrera. Además, creo que el sector del comercio electrónico irá creciendo poco a poco en México y el tener conocimientos sobre cómo desarrollar una tienda en línea, estamos un paso más adelante que los demás.

Durante el desarrollo de este proyecto, tuve varias dificultades tanto en la configuración del servidor como en el avance del mismo. Me di cuenta que a veces los errores no se encuentran directamente en el código de programación, sino en algún error de sintaxis de la Query. También, tuve un poco de dificultad en el manejo del diseño, ya que tuve aprender a manejar Bootstrap, pero al final no fue tan difícil.

Además, pude adquirir los conocimientos necesarios para utilizar el lenguaje de programación PHP y el manejo de una base de datos en el entorno PostgreSQL, ya que anteriormente venia de manejar MySQL. Al final, se cumplió el objetivo de realizar una tienda de artículos en línea, pero si faltó perfeccionar algunos módulos. En definitiva, creo que el trabajo ha sido muy duro durante estos días, pero el resultado ha merecido la pena, tanto por lo que es, y por las posibilidades futuras que tiene.

## 7. Enlaces consultados

- Reicek. (2015). ¿Qué es PostgreSQL y cuáles son sus ventajas? Platzi, 2019. Recuperado de: <https://platzi.com/blog/que-es-postgresql/>
- Bearnés, B. (2016). ¿Cómo configurar Virtual Host de Apache en Ubuntu 16,04? Community, 2019. Recuperado de: <https://www.digitalocean.com/community/tutorials/como-configurar-virtual-hosts-de-apache-en-ubuntu-16-04-es>
- Macagua. (2016). Instalación de phpPgAdmin en Debian. Wordpress, 2019. Recuperado de: <https://lcaballero.wordpress.com/2016/06/23/instalacion-de-phppgadmin-en-debian-jessie/>
- Bootstrap. (S/F). Introduction. Bootstrap, 2019. Recuperado de: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- Martínez, I. (2015). Conectarse mediante PDO con Mysql, PostgreSQL y Oracle. Formación Web Online, 2019. Recuperado de: <http://www.formacionwebonline.com/conectarse-mediante-pdo-con-mysql-postgresql-y-oracle/>
- Bootstrap. (S/F). Popovers. Bootstrap, 2019. Recuperado de: <https://getbootstrap.com/docs/4.0/components/popovers/>
- Develoteca. (2018). Carrito de compras Php, Mysql, PayPal. Develoteca, 2019. Recuperado de: [https://www.youtube.com/watch?v=tlkO9hhOqak&list=PLSuKjujFoGJ0XF\\_Gv0VpiTHxAtO7LL8jl](https://www.youtube.com/watch?v=tlkO9hhOqak&list=PLSuKjujFoGJ0XF_Gv0VpiTHxAtO7LL8jl)
- S/N. (2019). How to Install Visual Studio Code on Debian 9. Linuxize, 2019. Recuperado de: <https://linuxize.com/post/how-to-install-visual-studio-code-on-debian-9/>
- Zanna. (2017). How to run Visual Studio Code as root. Stack. Exchange, 2019. Recuperado de: <https://askubuntu.com/questions/803343/how-to-run-visual-studio-code-as-root>

- Mortensen, P. (2017). Apache error "Could not reliably determine the server's fully qualified domain name". Exchange, 2019. Recuperado de: <https://askubuntu.com/questions/256013/apache-error-could-not-reliably-determine-the-servers-fully-qualified-domain-n/256018>
- S/N. (2017). PostgreSQL PHP: Querying Data. PostgreSQL tutorial, 2019. Recuperado de: <http://www.postgresqltutorial.com/postgresql-php/query/>
- Fandi. (2017). Pdo\_pgsql dsn. PHP, 2019. Recuperado de: <https://www.php.net/manual/es/ref.pdo-pgsql.connection.php>
- Rodríguez, M. (2007). Fecha actual. Quick Links, 2019. Recuperado de: <https://www.postgresql.org/message-id/95335e4e0705290052y35b0fbb3vba77f2ea57db83ef@mail.gmail.com>
- Paypal. (S/F). Checkout. Paypal, 2019. Recuperado de: <https://developer.paypal.com/docs/archive/checkout/how-to/customize-button/#>
- Paypal. (S/F). Sandbox Accounts. Paypal, 2019. Recuperado de: <https://developer.paypal.com/developer/accounts/>

## 8. Anexos

- **Configuración Virtual Host:**

- **Estructura:**

- Raíz por defecto: /var/www/ProyectoFinalWebShop
    - Apache: /etc/apache2

- **Instalación:**

- Crear repositorio en GitHub.
    - Clonar repositorio: /var/www
    - Copiar y configurar archivo 000-default.conf a la carpeta clonada del repositorio.
    - Con el siguiente comando se copia:
      - cp/etc/apache2/sites-available/000-default.conf  
/etc/apache2/sites-available/<nombrearchivo>.conf
    - Habilitar el Virtual Host: a2ensite nombre\_repositorio.conf
    - Reiniciar el servicio: service postgresql restart
    - Configurar archivo /etc/hosts
      - Editar: 127.0.0.1 www.nombre.com.mx
    - Verificar en el navegador archivo index.php.

- **Bitácoras.**

- Error y Acceso.
    - Ubicación: /var/log/apache2
      - <nombreBitácora>-error.log
      - <nombreBitácora>-error-access.log

- **Configuración Base de datos:**

- **Creamos la base de datos:**

- CREATE USER admin\_shop WITH ENCRYPTED PASSWORD '<contraseña>';
    - CREATE DATABASE geekshop WITH OWNER admin\_shop;

- **Para conectarnos a la base de datos:**

- conecct: psql -h 127.0.0.1 -d geekshop -U < contraseña >



- **Código de la base de datos:**

```
CREATE TABLE usuarios (  
    idUsuario    serial        NOT NULL,  
    nombre       varchar (30)   NOT NULL,  
    aPaterno     varchar (30)   NOT NULL,  
    usuarioNombre varchar (30)   NOT NULL,  
    correo       varchar (60)   NOT NULL,  
    contraseña   varchar (32)   NOT NULL,  
    rol          varchar (10)    NULL,  
  
    CONSTRAINT pkUsuario  
    PRIMARY KEY(idUsuario),  
);
```

```
CREATE TABLE productos (  
    idProducto    serial        NOT NULL,  
    nombre       varchar (60)   NOT NULL,  
    precio        decimal (20,2) NOT NULL,  
    descripcion   TEXT          NOT NULL,  
    imagen        varchar (255) NOT NULL,  
  
    CONSTRAINT pkProducto  
    PRIMARY KEY(idProducto)  
);
```

```
CREATE TABLE ventas (  
    idVenta        serial        NOT NULL,  
    claveTransaccion varchar (250) NOT NULL,  
    datosP        TEXT          NOT NULL,  
    fecha         timestamp      NOT NULL,  
    correo        varchar (5000) NOT NULL,  
    total         decimal (60,2) NOT NULL,  
    status        varchar (200) NOT NULL,  
    usuarioNombre varchar (30)   NULL,  
  
    CONSTRAINT pkVenta  
    PRIMARY KEY(idVenta)  
);
```

```
CREATE TABLE detalleVenta (  
    idDetalleV    serial        NOT NULL,  
    idVenta        Int          NOT NULL,  
    idProducto    Int          NOT NULL,  
    precioUnitario decimal (20,2) NOT NULL,  
    cantidad      Int          NOT NULL,  
    descargado    Int          NOT NULL,
```

```
        usuario        varchar(30)  NULL,

        CONSTRAINT pkDetalle
        PRIMARY KEY(idDetalleV)
    );

ALTER TABLE detalleVenta
ADD CONSTRAINT fkDetalleVenta
FOREIGN KEY(idVenta)
REFERENCES ventas(idVenta)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE detalleVenta
ADD CONSTRAINT fkDetalleProducto
FOREIGN KEY(idProducto)
REFERENCES productos(idProducto)
ON DELETE CASCADE ON UPDATE CASCADE;
```

- **Repositorio GitHub:**

- Link donde se puede consultar los archivos utilizados en el proyecto "The Geek Store":
- <https://github.com/CesarDark/ProyectoFinalWebShop.git>